09 January 2022

Machine Learning Project

PGP-DSBA June-Batch

SOOREJ KUNNOOL BALAN

2022

Table of Contents

[1. Problem 1 3](#_Toc92662107)

[1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it. 4](#_Toc92662108)

[Data Description 4](#_Toc92662109)

[Sample of the dataset: 4](#_Toc92662110)

[Concise Data Summary: 4](#_Toc92662111)

[1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers. 7](#_Toc92662112)

[1.3 Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30). 21](#_Toc92662113)

[1.4 Apply Logistic Regression and LDA (linear discriminant analysis). 22](#_Toc92662114)

[1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results. 27](#_Toc92662115)

[1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging) and Boosting. 31](#_Toc92662116)

[1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized. 36](#_Toc92662117)

[1.8 Based on these predictions, what are the insights? 41](#_Toc92662118)

[2. Problem 2 43](#_Toc92662119)

[2.1 Find the number of characters, words, and sentences for the mentioned documents. 43](#_Toc92662120)

[2.2 Remove all the stopwords from all three speeches. 43](#_Toc92662121)

[2.3 Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. 44](#_Toc92662122)

[2.4 Plot the word cloud of each of the three speeches. (after removing the stopwords). 44](#_Toc92662123)

[THE END 45](#_Toc92662124)

List of Tables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table-1.1 Dataset Sample | 4 |  | Table-1.46 Classification Report: LDA – Training Data | 38 |
| Table-1.2 Column renamed Dataset Sample | 4 |  | Table-1.47 Confusion Matrix: LDA– Testing Data | 38 |
| Table-1.3 Concise data summary | 5 |  | Table-1.48 Confusion Matrix: KNN – Training Data | 38 |
| Table-1.4 Duplicate records | 5 |  | Table-1.49 Confusion Matrix: KNN– Testing Data | 38 |
| Table-1.5 Data Summary of numeric columns | 6 |  | Table-1.50 Classification Report: KNN– Training Data | 38 |
| Table-1.6 Data Summary of object columns | 6 |  | Table-1.51 Confusion Matrix: KNN– Testing Data | 38 |
| Table-1.7 Skewness | 6 |  | Table-1.52 Confusion Matrix: NB– Training Data | 38 |
| Table-1.8 vote: value count | 12 |  | Table-1.53 Confusion Matrix: NB– Testing Data | 38 |
| Table-1.9 gender: value count | 13 |  | Table-1.54 Classification Report: NB– Training Data | 39 |
| Table-1.10 economic\_cond\_national/household: value count | 14 |  | Table-1.55 Confusion Matrix: NB – Testing Data | 39 |
| Table-1.11 Correlation matrix | 14 |  | Table-1.56 Confusion Matrix: BAG – Training Data | 39 |
| Table-1.12 Range of numeric columns | 21 |  | Table-1.57 Confusion Matrix: BAG– Testing Data | 39 |
| Table-1.13 Data summary of scaled dataset | 21 |  | Table-1.58 Classification Report: BAG– Training Data | 40 |
| Table-1.14 Scaled dataset sample | 21 |  | Table-1.59 Confusion Matrix: BAG– Testing Data | 40 |
| Table-1.15 Scaled and encoded dataset sample | 22 |  | Table-1.60 Confusion Matrix: ADB – Training Data | 40 |
| Table-1.16 Train and test predictor dataset concise summary | 22 |  | Table-1.61 Confusion Matrix: ADB– Testing Data | 40 |
| Table-1.17 Value counts of training/test target variable | 22 |  | Table-1.62 Classification Report: ADB– Training Data | 40 |
| Table-1.18 Confusion Matrix: Logistic Regression – Training Data | 23 |  | Table-1.63 Confusion Matrix: ADB– Testing Data | 40 |
| Table-1.19 Classification Report: Logistic Regression – Training Data | 23 |  | Table-1.64 Confusion Matrix: GDB – Training Data | 40 |
| Table-1.20 Confusion Matrix: Logistic Regression – Testing Data | 24 |  | Table-1.65 Confusion Matrix: GDB– Testing Data | 40 |
| Table-1.21 Classification Report: Logistic Regression – Testing Data | 24 |  | Table-1.66 Classification Report: GDB– Training Data | 41 |
| Table-1.22 Confusion Matrix: LDA – Training Data | 25 |  | Table-1.67 Confusion Matrix: GDB– Testing Data | 41 |
| Table-1.23 Classification Report: LDA – Training Data | 25 |  | Table-1.68 Model comparison | 41 |
| Table-1.24 Confusion Matrix: LDA – Testing Data | 25 |  | Table-2.1 Stop words: Sample | 43 |
| Table-1.25 Classification Report: LDA – Testing Data | 26 |  | Table-2.2 Stop words statistics | 43 |
| Table-1.25 Model comparison: LGR vs LDA | 26 |  | Table-2.3 Speech statistics after stop words removal | 44 |
| Table-1.26 Confusion Matrix: KNN – Training Data | 27 |  | Table-2.4 Top 3 words of each speech | 44 |
| Table-1.27 Classification Report: KNN – Training Data | 28 |  |  |  |
| Table-1.28 Confusion Matrix: KNN – Testing Data | 28 |  |  |  |
| Table-1.29 Classification Report: KNN – Testing Data | 28 |  |  |  |
| Table-1.30 Confusion Matrix: NB – Training Data | 29 |  |  |  |
| Table-1.31 Classification Report: NB – Training Data | 29 |  |  |  |
| Table-1.32 Confusion Matrix: NB – Testing Data | 30 |  |  |  |
| Table-1.33 Classification Report: NB – Testing Data | 30 |  |  |  |
| Table-1.34 Model Comparison : KNN vs NB | 30 |  |  |  |
| Table-1.35 Feature Importance – Logistic Regression | 32 |  |  |  |
| Table-1.36 Feature Importance – LDA | 32 |  |  |  |
| Table-1.37 Feature Importance – Bagging | 34 |  |  |  |
| Table-1.38 Feature Importance – Adaboosting | 35 |  |  |  |
| Table-1.39 Feature Importance – Gradient boosting | 36 |  |  |  |
| Table-1.40 Confusion Matrix: Logistic Regression – Training Data | 36 |  |  |  |
| Table-1.41 Confusion Matrix: Logistic Regression – Testing Data | 36 |  |  |  |
| Table-1.42 Classification Report: LGR – Training Data | 37 |  |  |  |
| Table-1.43 Confusion Matrix: LGR– Testing Data | 37 |  |  |  |
| Table-1.44 Confusion Matrix: LDA – Training Data | 37 |  |  |  |
| Table-1.45 Confusion Matrix: LDA – Testing Data | 37 |  |  |  |

List of Figures

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Figure-1.1 Histogram & Boxplot : age | 7 |  | Figure-1.48 ROC Curve: GDB– Training Data | | 41 |
| Figure-1.2 Histogram & Boxplot: economic\_cond\_national | 8 |  | Figure-1.49 ROC Curve: GDB – Testing Data | | 41 |
| Figure-1.3 Histogram & Boxplot : economic\_cond\_household | 9 |  | Figure-2.1 Word cloud: Roosevelt’s speech | | 44 |
| Figure-1.4 Histogram & Boxplot : Blair | 10 |  | Figure-2.2 Word cloud: Kennedy’s speech | | 44 |
| Figure-1.5 Histogram & Boxplot : Hague | 10 |  | Figure-2.3 Word cloud: Nixon’s speech | | 45 |
| Figure-1.6 Histogram & Boxplot : Europe | 11 |  |  | |  |
| Figure-1.7 Histogram & Boxplot : political\_knowledge | 12 |  |  | |  |
| Figure-1.8 Count-plot & Pie-chart : vote | 13 |  |  | |  |
| Figure-1.9 Count-plot & Pie-chart : gender | 13 |  |  | |  |
| Figure-1.11 Pairplot | 15 |  |  | |  |
| Figure-1.12 Countplot: economic\_cond\_household, hue-vote | 16 |  |  | |  |
| Figure-1.13 Countplot: economic\_cond\_national, hue-vote | 16 |  |  | |  |
| Figure-1.14 Countplot: political\_knowledge, hue-vote | 16 |  |  | |  |
| Figure-1.15 Countplot: gender, hue-vote | 17 |  |  | |  |
| Figure-1.16 Countplot: Europe, hue-vote | 17 |  |  | |  |
| Figure-1.17 Boxplot: Blair vs age | 17 |  |  | |  |
| Figure-1.18 Boxplot: Hague vs age | 18 |  |  | |  |
| Figure-1.19 Countplot: Blair, hue-vote | 18 |  |  | |  |
| Figure-1.20 Countplot: Hague, hue-vote | 18 |  |  | |  |
| Figure-1.21 Countplot: gender, hue-Blair | 19 |  |  | |  |
| Figure-1.22 Countplot: gender, hue-Hague | 19 |  |  | |  |
| Figure-1.23 Countplot: economic\_cond\_household, hue-Blair | 19 |  |  | |  |
| Figure-1.24 Countplot: economic\_cond\_household, hue-Hague | 20 |  |  | |  |
| Figure-1.25 Countplot: economic\_cond\_household, hue-Europe | 20 |  |  | |  |
| Figure-1.26 Countplot: economic\_cond\_national, hue-Europe | 20 |  |  | |  |
| Figure-1.27 ROC Curve: Logistic Regression – Training Data | 23 |  |  |  | |
| Figure-1.28 ROC Curve: Logistic Regression – Testing Data | 24 |  |  |  | |
| Figure-1.29 ROC Curve: LDA – Training Data | 25 |  |  |  | |
| Figure-1.30 ROC Curve: LDA – Testing Data | 26 |  |  |  | |
| Figure-1.31 KNN: mce vs k | 27 |  |  |  | |
| Figure-1.32 ROC Curve: KNN – Training Data | 27 |  |  |  | |
| Figure-1.33 ROC Curve: KNN – Testing Data | 28 |  |  |  | |
| Figure-1.34 ROC Curve: NB – Training Data | 29 |  |  |  | |
| Figure-1.35 ROC Curve: NB – Testing Data | 30 |  |  |  | |
| Figure-1.36 ROC Curve: LGR – Training Data | 37 |  |  |  | |
| Figure-1.37 ROC Curve: LGR – Testing Data | 37 |  |  |  | |
| Figure-1.38 ROC Curve: LDA – Training Data | 37 |  |  |  | |
| Figure-1.39 ROC Curve: LDA – Testing Data | 37 |  |  |  | |
| Figure-1.40 ROC Curve: KNN– Training Data | 38 |  |  |  | |
| Figure-1.41 ROC Curve: KNN – Testing Data | 38 |  |  |  | |
| Figure-1.42 ROC Curve: NB – Training Data | 39 |  |  |  | |
| Figure-1.43 ROC Curve: NB – Testing Data | 39 |  |  |  | |
| Figure-1.44 ROC Curve: BAG– Training Data | 39 |  |  |  | |
| Figure-1.45 ROC Curve: BAG – Testing Data | 39 |  |  |  | |
| Figure-1.46 ROC Curve: ADB– Training Data | 40 |  |  |  | |
| Figure-1.47 ROC Curve: ADB – Testing Data | 40 |  |  |  | |

# Problem 1

You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.



## Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.

### Data Description

1. vote : Party choice. Values present are conservative and labour.
2. age : Age of the voter in years (continuous variable).
3. economic.cond.national: Assessment of current national economic conditions, 1 to 5 (ordinal variable).
4. economic.cond.household: Assessment of current household economic conditions, 1 to 5 (ordinal variable).
5. Blair: Assessment of the Labour leader, 1 to 5 (ordinal variable).
6. Hague: Assessment of the Conservative leader, 1 to 5 (ordinal variable).
7. Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment (ordinal variable).
8. political.knowledge: Knowledge of parties' positions on European integration, 0 to 3 (ordinal variable).
9. gender: Gender of the voter. Values present are female or male.

### Sample of the dataset:

We can remove the first column in the csv file, which is a serial number column. The dataset sample is as shown below:

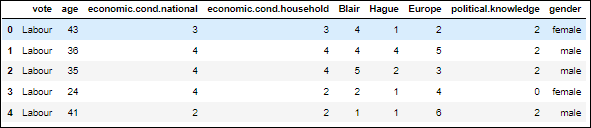


Table-1.1 Dataset Sample

The columns names have ‘.’ which could interfere with python commands hence we will replace ‘.’ In column names with ‘\_’ as shown below:

Table

Description automatically generated

Table-1.2 Column renamed Dataset Sample

### Concise Data Summary:

Lets look at the concise data summary of the dataset.

Text

Description automatically generated

Table-1.3 Concise data summary

#### Let us check the type of variables in the data frame

There are a total of 1525 observations and 9 columns in the dataset. We have 2 columns of object type and rest 7 of int64 type.

#### Check for missing values in the dataset

From Table-1.3 we can see that all the columns have 1525 non-null values. Hence there are no missing/null values in the dataset.

#### Check for duplicate observations in the dataset

There are 8 duplicate records as shown below:

A picture containing application

Description automatically generated

Table-1.4 Duplicate records

Since we don’t have any unique identifier of the voter, the above 8 records could indeed represent data of different individuals than actual duplicates. Since the count 8 is miniscule compared to the sample 1525 observations, let’s not delete the duplicates.

#### Data summary

Graphical user interface, text

Description automatically generated

Table-1.5 Data Summary of numeric columns

The above summary table of numeric columns provide statistical information such as count, mean, standard deviation, quartile values and dataset min, max values.

A screenshot of a computer

Description automatically generated with medium confidence

Table-1.6 Data Summary of object columns

The above summary table of categorical columns provide statistical information such as count, how many unique values are present, and the frequency of maximum occurring value.

We will be delving deeper into these while performing univariate analysis in 1.2

#### Skewness

The below table shows the skewness of the numerical columns in the dataset:

Text

Description automatically generated

Table-1.7 Skewness

Skewness is the measure of how much the probability distribution of a random variable deviates from the normal distribution. Here we can see that except for ‘Blair’ all other numeric columns have skewness between -0.5 to 0.5, hence we can say that the distributions for these are approximately symmetric.

Blair has a value -0.53, and is moderately skewed.

Only ‘age’ and ‘Hague’ are positively skewed, rest all variables are negatively skewed.

## Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

#### Univariate Analysis:

Let’s check the central measures of tendency, quartiles, histogram, and boxplot of all 7 continuous columns.

1. age

Age of voter’s is a continuous variable with the below stats (refer Table 1.5):

Mean = 54.182295

Standard Deviation = 15.711209

Min value in dataset = 24

Max value in dataset = 93

Range = Min – Max = 69

Q1(1st Quartile) = 41

Q2(2nd Quartile)/Median = 53

Q3(3rd Quartile) = 67

IQR(Inter-Quartile Range) = Q3- Q1 = 26

Quartile Min value = max(Qmin,Q1 – 1.5 \* IQR) = 24

Quartile Max value = min(Qmax,Q3 + 1.5 \* IQR) = 93

Chart, histogram

Description automatically generated

Figure-1.1 Histogram & Boxplot : age

Figure-1.1 depicts the histogram and boxplot of “age” which shows positive skewness in the data. The data depicts the presence of multi-modes as we can see multiple peaks in the histogram.

From the boxplot we can see that there are no outliers present in ‘age’.

1. economic\_cond\_national

Assessment of current national economic conditions is a continuous variable (ordinal) with the below stats (refer Table 1.5):

Mean = 3.245902

Standard Deviation = 0.880969

Min value in dataset = 1

Max value in dataset = 5

Range = Min – Max = 4

Q1(1st Quartile) = 3

Q2(2nd Quartile)/Median = 3

Q3(3rd Quartile) = 4

IQR(Inter-Quartile Range) = Q3- Q1 = 1

Quartile Min value = max(Qmin,Q1 – 1.5 \* IQR) = 2

Quartile Max value = min(Qmax,Q3 + 1.5 \* IQR) = 5

Chart, histogram, box and whisker chart

Description automatically generated

Figure-1.2 Histogram & Boxplot: economic\_cond\_national

Figure-1.2 depicts the histogram and boxplot of “economic\_cond\_national” which shows negativeskewness in the data. The data depicts the presence of multi-modes as we can see multiple peaks in the histogram.

From the boxplot we can see that there are outliers present in ‘economic\_cond\_national’. The quartile minimum value is 2 and we have values 1. We will discuss this further in outlier treatment section.

1. economic\_cond\_household

Assessment of current household economic conditions, is a continuous variable (ordinal) with the below stats (refer Table 1.5):

Mean = 3.140328

Standard Deviation = 0.929951

Min value in dataset = 1

Max value in dataset = 5

Range = Min – Max = 4

Q1(1st Quartile) = 3

Q2(2nd Quartile)/Median = 3

Q3(3rd Quartile) = 4

IQR(Inter-Quartile Range) = Q3- Q1 = 1

Quartile Min value = max(Qmin,Q1 – 1.5 \* IQR) = 2

Quartile Max value = min(Qmax,Q3 + 1.5 \* IQR) = 5

Chart, histogram, box and whisker chart

Description automatically generated

Figure-1.3 Histogram & Boxplot : economic\_cond\_household

Figure-1.3 depicts the histogram and boxplot of “economic\_cond\_household” which shows negative skewness in the data. The data depicts the presence of multi-modes as we can see multiple peaks in the histogram.

From the boxplot we can see that there are outliers present in ‘economic\_cond\_household’. The quartile minimum value is 2 and we have values 1. We will discuss this further in outlier treatment section.

1. Blair

Assessment of the Labour leader is a continuous variable (ordinal) with the below stats (refer Table 1.5):

Mean = 3.334426

Standard Deviation = 1.174824

Min value in dataset = 1

Max value in dataset = 5

Range = Min – Max = 4

Q1(1st Quartile) = 2

Q2(2nd Quartile)/Median = 4

Q3(3rd Quartile) = 4

IQR(Inter-Quartile Range) = Q3- Q1 = 2

Quartile Min value = max(Qmin,Q1 – 1.5 \* IQR) = 1

Quartile Max value = min(Qmax,Q3 + 1.5 \* IQR) = 5

Chart, histogram

Description automatically generated

Figure-1.4 Histogram & Boxplot : Blair

Figure-1.4 depicts the histogram and boxplot of “Blair” which shows negative skewness in the data. The data depicts the presence of multi-modes as we can see multiple peaks in the histogram.

From the boxplot we can see that there are no outliers present in ‘Blair’.

1. Hague

Assessment of the Conservative leader is a continuous variable (ordinal) with the below stats (refer Table 1.5):

Mean = 2.746885

Standard Deviation = 1.230703

Min value in dataset = 1

Max value in dataset = 5

Range = Min – Max = 4

Q1(1st Quartile) = 2

Q2(2nd Quartile)/Median = 2

Q3(3rd Quartile) = 4

IQR(Inter-Quartile Range) = Q3- Q1 = 2

Quartile Min value = max(Qmin,Q1 – 1.5 \* IQR) = 1

Quartile Max value = min(Qmax,Q3 + 1.5 \* IQR) = 5

Chart, histogram

Description automatically generated

Figure-1.5 Histogram & Boxplot : Hague

Figure-1.5 depicts the histogram and boxplot of “Hague” which shows positive skewness in the data. The data depicts the presence of multi-modes as we can see multiple peaks in the histogram.

From the boxplot we can see that there are no outliers present in ‘Hague’.

1. Europe

A 11-point scale that measures respondents' attitudes toward European integration is a continuous variable (ordinal) with the below stats (refer Table 1.5):

Mean = 6.728525

Standard Deviation = 3.297538

Min value in dataset = 1

Max value in dataset = 11

Range = Min – Max = 10

Q1(1st Quartile) = 4

Q2(2nd Quartile)/Median = 6

Q3(3rd Quartile) = 10

IQR(Inter-Quartile Range) = Q3- Q1 = 6

Quartile Min value = max(Qmin,Q1 – 1.5 \* IQR) = 1

Quartile Max value = min(Qmax,Q3 + 1.5 \* IQR) = 11

Chart, histogram

Description automatically generated

Figure-1.6 Histogram & Boxplot : Europe

Figure-1.6 depicts the histogram and boxplot of “Europe” which shows negative skewness in the data. The data depicts the presence of multi-modes as we can see multiple peaks in the histogram.

From the boxplot we can see that there are no outliers present in ‘Europe’.

1. political\_knowledge

Knowledge of parties' positions on European integration is a continuous variable (ordinal) with the below stats (refer Table 1.5):

Mean = 1.542295

Standard Deviation = 1.083315

Min value in dataset = 0

Max value in dataset = 3

Range = Min – Max = 3

Q1(1st Quartile) = 0

Q2(2nd Quartile)/Median = 2

Q3(3rd Quartile) = 2

IQR(Inter-Quartile Range) = Q3- Q1 = 2

Quartile Min value = max(Qmin,Q1 – 1.5 \* IQR) = 0

Quartile Max value = min(Qmax,Q3 + 1.5 \* IQR) = 3

Chart, histogram

Description automatically generated

Figure-1.7 Histogram & Boxplot : political\_knowledge

Figure-1.7 depicts the histogram and boxplot of “political\_knowledge” which shows negative skewness in the data. The data depicts the presence of multi-modes as we can see multiple peaks in the histogram.

From the boxplot we can see that there are no outliers present in ‘political\_knowledge’.

Let’s look at the categorical variables:

1. vote

Party choice. The below table shows the count of values in vote column.

Text

Description automatically generated

Table-1.8 vote: value count

Let’s look at the count-plot for vote variable and the pie-chart for % distribution.

Chart

Description automatically generated

Figure-1.8 Count-plot & Pie-chart : vote

We can see that vote is higher for Labour party at 69.7%(1063) against Conservative party, 30.3% (462) of the total 1525 observations.

1. gender

Gender of the voter. The below table shows the count of values in gender column.

Graphical user interface, text

Description automatically generated with medium confidence

Table-1.9 gender: value count

Let’s look at the count-plot for gender variable and the pie-chart for % distribution.

Chart, bar chart, pie chart

Description automatically generated

Figure-1.9 Count-plot & Pie-chart : gender

We can see that female voters are higher at 53.25%(812 in number) against male voters, 46.75% (713) of the total 1525 observations.

#### Outlier Treatment:

We have seen outliers present in 2 variables namely economic\_cond\_national (Figure 1.2) and economic\_cond\_household (Figure 1.3). Both the features have outlier value at 1, where the quartile minimum is 2.

Let’s look at the percent of observations.

Shape

Description automatically generated Shape

Description automatically generated

Table-1.10 economic\_cond\_national/economic\_cond\_household: value count

We can see that the percentage of outliers (value 1) is less than 5% in both feature and this is not an erroneous data. Hence we will not be treating these outliers.

#### Bivariate/Multivariate Analysis:

Let us plot a heat map for the correlation matrix of given data frame.

Chart, waterfall chart

Description automatically generated

Table-1.11 Correlation matrix

We can see moderate positive correlation between the following variables:

* Blair and economic\_cond\_national
* Blair and economic\_cond\_household
* economic\_cond\_national and economic\_cond\_household
* Europe and Hague

We can see moderate negative correlation between the following variables:

* Blair and Hague

All other correlations are weak.

Since the overall correlation is less between variables, we should not be facing issues arising due to multicollinearity.

Let’s also check the Pairplot:

Chart

Description automatically generated

Figure-1.11 Pairplot

Pairplot substantiates the heatmap and displays the correlation between variables as stated above.

Let us also look at interaction of variables and check for inferences.

* economic\_cond\_household & vote

From Figure 1.12, we can see that irrespective of the economic condition of the voter’s household, majority of voters have voted for the Labour party. The vote % between Labour and Conservative party is comparable for 1 and 2 economic condition household, but for 3,4 and 5 Labour party enjoys a clear majority with 4 having the maximum vote difference.

Chart, bar chart

Description automatically generated

Figure-1.12 Countplot: economic\_cond\_household, hue-vote

* economic\_cond\_national & vote

From Figure 1.13, we can see that majority of the voters who have higher assessment of current national economic condition (3,4 and 5) have voted for Labour party, but majority of those voters who had lower assessment (1 and 2) have voted for the Conservative party. The lead enjoyed by Conservative party is minimal for 1 and 2, whereas the majority enjoyed by Labour party for voters assessing current national economic condition (3,4 and 5) is at least 100% or more.

Chart, bar chart

Description automatically generated

Figure-1.13 Countplot: economic\_cond\_national, hue-vote

* political\_knowledge & vote

From Figure 1.14, we can see that irrespective of the voter’s knowledge of the position of the parties on European integration, majority of the voters in each class have voted for the Labour party and the party enjoys a comfortable margin over Conservative party at least 70% or more.

Chart, bar chart

Description automatically generated

Figure-1.14 Countplot: political\_knowledge, hue-vote

* gender & vote

From Figure 1.15, we can see that irrespective of the voter’s gender, Labour party enjoys a clear majority over Conservative party at least 100% or more.

Chart, bar chart, treemap chart

Description automatically generated

Figure-1.15 Countplot: gender, hue-vote

* Europe & vote

From Figure 1.16, we can see that Conservative party enjoys a majority among voters with high Eurosceptic sentiment (9,10 and 11). For all other assessment values Labor party enjoys a clear majority. We can see that the share of voters for Conservative party seems to be increasing with Eurosceptic sentiment, whereas it seems that for voters who have chosen Labour party seems not to consider European integration as a factor.

Chart, bar chart

Description automatically generated

Figure-1.16 Countplot: Europe, hue-vote

* Blair and age

From Figure 1.17, we can see that on an average with increase in age of the voter, the assessment of the Labour leader Blair seems to increase.

Chart, box and whisker chart

Description automatically generated

Figure-1.17 Boxplot: Blair vs age

* Hague and age

From Figure 1.18, we can see that age of the voter does not seem to be a factor influencing the assessment of the Conservative leader Hague.

Chart, box and whisker chart

Description automatically generated

Figure-1.18 Boxplot: Hague vs age

* Blair and vote

From Figure 1.19, we can see that voters who have lower assessments for Blair (1 and 2) have voted for Conservative party. But interestingly voters do not seem to have a neutral stand towards Blaire (3). Majority of voters who have high assessment of Blair (4 and 5) have voted for Labour party.

Chart, bar chart

Description automatically generated

Figure-1.19 Countplot: Blair, hue-vote

* Hague and vote

From Figure 1.20, we can see that , but for 4 the majority is not that high.

Chart, bar chart

Description automatically generated

Figure-1.20 Countplot: Hague, hue-vote

* gender and Blair

From Figure 1.21, we can see that irrespective of the gender, majority of the voters have a higher assessment of Blair (4 and 5).

Chart, bar chart

Description automatically generated

Figure-1.21 Countplot: gender, hue-Blair

* gender and Hague

From Figure 1.22, we can see that irrespective of the gender, majority of the voters have a lower assessment of Hague(1 and 2).

Chart, bar chart

Description automatically generated

Figure-1.22 Countplot: gender, hue-Hague

* economic\_cond\_household and Blair

From Figure 1.23, we can see that majority of voters who assessed economic\_cond\_household as 3 and above (which is majority of the sample) have assessed Blaire highly (4 and 5).

Chart, bar chart

Description automatically generated

Figure-1.23 Countplot: economic\_cond\_household, hue-Blair

* economic\_cond\_household and Hague

From Figure 1.24, we can see that majority of voters who assessed economic\_cond\_household as 3 and above (which is majority of the sample) have assessed Hague low (1 and 2), whereas below 3 we can see that majority of voters have assessed Hague highly ( 4and 5).

Chart, bar chart

Description automatically generated

Figure-1.24 Countplot: economic\_cond\_household, hue-Hague

* economic\_cond\_household and Europe

From Figure 1.25, we can see that irrespective of the household economic condition majority of the voters are Eurosceptic (9,10 and 11).

Chart, bar chart

Description automatically generated

Figure-1.25 Countplot: economic\_cond\_household, hue-Europe

* economic\_cond\_national and Europe

From Figure 1.25, we can see that irrespective of the national economic condition assessment, majority of the voters in each class are Eurosceptic (9,10 and 11).

Chart, bar chart

Description automatically generated

Figure-1.26 Countplot: economic\_cond\_national, hue-Europe

## Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30).

Lets check the range of numeric columns in the dataset:

Text

Description automatically generated

Table-1.12 Range of numeric columns

From table 1.12 we can see that the range of data varies from 3 to 69. Hence it’s a good idea to scale the data, else in distance-based machine learning algorithms such as KNN the feature with higher range (accounting for different attributes), in this scenario age will have undue bias on the model. We will use standard scaler and normalize the numerical columns in the dataset, as we have seen that the histogram of numerical columns in the univariate section, generally was gaussian in nature. Lets look at the data summary of the scaled numeric dataset:

Graphical user interface, application

Description automatically generated

Table-1.13 Data summary of scaled dataset

We can see that the dataset has been normalized with mean approaching 0 and standard deviation near to 1. Lets look at the head of scaled dataset.

Graphical user interface, application

Description automatically generated

Table-1.14 Scaled dataset sample

Next we will encode the 2 object variables, vote and gender. Since the features are not ordinal in nature we can proceed with dummy encoding.

Lets look at the dataset after dummy encoding:

Graphical user interface, application

Description automatically generated with medium confidence

Table-1.15 Scaled and encoded dataset sample

So the 2 dummy encoded features are gender\_male (if value is 1 its male, else female) and vote\_Labour(if value is 1, vote is for Labour else for Conservative).

In the above dataset, vote\_Labour is our target variable, and all other features are predictors.

We will split the above data into test and train, predictor and target variable sets, with the split of 70% of observations for training data set and rest 30% for testing.

Table

Description automatically generated Text

Description automatically generated

Table-1.16 Train and test predictor dataset concise summary

Similarly, the target train and test have 1067 and 458 entries respectively.

Let’s look at the distribution of values in the training and test target variable sets.

Text

Description automatically generated Text

Description automatically generated

Table-1.17 Value counts of training/test target variable

From figure 1.17, we can see that the distribution of target variable in both training and dataset have similar percentages. Hence we can expect the model training and subsequent prediction on test dataset, to have enough observations to make reasonable predictions. If the training set had too little observations of any one value, it will cause overfitting and perform abysmally against testing data.

## Apply Logistic Regression and LDA (linear discriminant analysis).

#### Logistic Regression:

Let’s first look at the hyper-parameters:

* solver – Since our training data set has about 1000 observations(small dataset) and the target variable is binary, liblinear will be an ideal choice.
* penalty – liblinear supports l1 penalty, hence we will use that.

We will let other hyperparameters such as tolerance(0.0001) and max\_iteration(100) to be at default value and check in case if convergence does not happen.

Let us look at the model performance on training data:

* The accuracy of the logistic regression model on training data is 0.8406747891283973.
* Confusion matrix is given below:

Text

Description automatically generated

Table-1.18 Confusion Matrix: Logistic Regression – Training Data

* The ROC\_AUC score for Logistic Regression model on training data is 0.889511105647078.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.27 ROC Curve: Logistic Regression – Training Data

* The classification report is given below:

Table, calendar

Description automatically generated

Table-1.19 Classification Report: Logistic Regression – Training Data

The model has an accuracy of 84% on the testing data and the ROC-AUC score of 0.8895 shows that the model is performing well.

Lets look at the model’s performance on the testing data:

* The accuracy of Logistic regression model on testing set is 0.8231441048034934.
* Confusion matrix is given below:

Text

Description automatically generated with medium confidence

Table-1.20 Confusion Matrix: Logistic Regression – Testing Data

* The ROC\_AUC score for Logistic Regression model on testing data is 0.8828330206378986.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.28 ROC Curve: Logistic Regression – Testing Data

* The classification report is given below:

Table

Description automatically generated

Table-1.21 Classification Report: Logistic Regression – Testing Data

The model has an accuracy of 82% on the testing data and the ROC-AUC score of 0.8828 shows that the model is performing well. The precision levels are at 87% and we can conclude that Logistic Regression model is performing well against the data. There is slight over-fitting as the accuracy for train dataset is slightly higher than test dataset.

#### Linear Discriminant Analysis (LDA):

Let’s first look at the hyper-parameters:

* solver – We will start with the default solver svd- Singular value decomposition.Since our training data set has about 1000 observations(small dataset) and the target variable is binary, liblinear will be an ideal choice.
* tolerance –we will start with default of 0.0001.

Let us look at the model performance on training data:

* The accuracy of LDA model on training data is 0.8369259606373008.
* Confusion matrix is given below:



Table-1.22 Confusion Matrix: LDA – Training Data

* The ROC\_AUC score for LDA model on training data is 0.8892242439144332.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.29 ROC Curve: LDA – Training Data

* The classification report is given below:

A screenshot of a computer

Description automatically generated with medium confidence

Table-1.23 Classification Report: LDA – Training Data

The model has an accuracy of 84% on the training data and the ROC-AUC score of 0.8892 shows that the model is performing well.

Let’s look at the model’s performance on the testing data:

* The accuracy of LDA model on testing set is 0.8187772925764192.
* Confusion matrix is given below:

Text

Description automatically generated

Table-1.24 Confusion Matrix: LDA – Testing Data

* The ROC\_AUC score for LDA model on testing data is 0.8837711069418386.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.30 ROC Curve: LDA – Testing Data

* The classification report is given below:

A screenshot of a computer

Description automatically generated with medium confidence

Table-1.25 Classification Report: LDA – Testing Data

The model has an accuracy of 82% on the testing data and the ROC-AUC score of 0.8838 shows that the model is performing well. The precision levels are at 87% and we can conclude that LDA model is performing well against the data.

Lets compare the Logistic regression and LDA model’s performance.

Table

Description automatically generated

Table-1.25 Model comparison: LGR vs LDA

The logistic regression model and linear discriminant analysis models have very comparable performance. Both models are performing aptly with very slight overfitting. The logistic regression model’s overall accuracy, recall, roc\_auc\_score and f1-score are higher than that of LDA.

Hence we will consider logistic regression as the preferred model.

## Apply KNN Model and Naïve Bayes Model. Interpret the results.

#### K Nearest Neighbor (KNN):

Let us start with the main hyper-parameters of this model.

* n\_neighbors: number of neighbors to be considered.
* weights : since our dataset is scaled we can consider uniform (which is default).
* Algorithm: we will use auto (which is default) so that the model chooses the appropriate algorithm.

To determine the k value, we will run the KNN model for different k values and see which has the lowest mis-classification error (1- model score for the test dataset). On plotting the mce values for various k we get the below graph:

Chart, line chart

Description automatically generated

Figure-1.31 KNN: mce vs k

We can see that mce is lowest for value of k as 5, hence we will model based on 5 n\_neighbor.

Let us look at the model performance on training data:

* The accuracy of KNN model on training data is 0.8622305529522024.
* Confusion matrix is given below:

Text

Description automatically generated

Table-1.26 Confusion Matrix: KNN – Training Data

* The ROC\_AUC score for KNN model on training data is 0.9300713056306861.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.32 ROC Curve: KNN – Training Data

* The classification report is given below:

A picture containing table

Description automatically generated

Table-1.27 Classification Report: KNN – Training Data

The model has an accuracy of 86% on the training data and the ROC-AUC score of 0.93 shows that the model is performing well.

Let’s look at the model’s performance on the testing data:

* The accuracy of KNN model on testing set is 0.8275109170305677.
* Confusion matrix is given below:

A picture containing text

Description automatically generated

Table-1.28 Confusion Matrix: KNN – Testing Data

* The ROC\_AUC score for KNN model on testing data is 0.8678939962476547.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.33 ROC Curve: KNN – Testing Data

* The classification report is given below:

Table

Description automatically generated with medium confidence

Table-1.29 Classification Report: KNN – Testing Data

The model has an accuracy of 83% on the testing data and the ROC-AUC score of 0.8679 shows that the model is performing well. The precision levels are at 88% and we can conclude that KNN model is performing well against the data and there is slight over-fitting as accuracy for training data set is slightly higher than test dataset.

#### Naïve Bayes (NB):

We will implement NB model via GaussianNB implementation of sklearn library’s naïve\_bayes library.

We will run the model without passing any hyper parameters, such as priors (prior probabilities of classes, default is none) or var\_smoothing(added to variances of features for calculation stability, default is 1e-9)

Let us look at the model performance on training data:

* The accuracy of NB model on training data is 0.8331771321462043.
* Confusion matrix is given below:

Text

Description automatically generated

Table-1.30 Confusion Matrix: NB – Training Data

* The ROC\_AUC score for NB model on training data is 0.8864703712810426.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.34 ROC Curve: NB – Training Data

* The classification report is given below:

Calendar

Description automatically generated with medium confidence

Table-1.31 Classification Report: NB – Training Data

The model has an accuracy of 83% on the training data and the ROC-AUC score of 0.8865 shows that the model is performing well.

Let’s look at the model’s performance on the testing data:

* The accuracy of NB model on testing set is 0.8253275109170306.
* Confusion matrix is given below:



Table-1.32 Confusion Matrix: NB – Testing Data

* The ROC\_AUC score for KNN model on testing data is 0.8845450281425891.
* The ROC curve is given below:

Chart, line chart

Description automatically generated

Figure-1.35 ROC Curve: NB – Testing Data

* The classification report is given below:

A screenshot of a computer

Description automatically generated with medium confidence

Table-1.33 Classification Report: NB – Testing Data

The model has an accuracy of 83% on the testing data and the ROC-AUC score of 0.8845 shows that the model is performing well. The precision levels are at 89% and we can conclude that NB model is performing well against the data and there is very slight overfitting, as accuracy is higher for train dataset than test.

Lets compare the performance of both the models – KNN and NB.

A screenshot of a computer

Description automatically generated

Table-1.34 Model Comparison : KNN vs NB

Between KNN and NB when we compare, we can see that even though KNN model has higher scores than NB on the training dataset, the KNN’s testing dataset has performed much poorly (when compared to KNN training) than that in the case of NB. Looks like KNN has a slight more over-fitting issue as compared to NB. If these models are run on larger datasets, its potentially possible that KNN will perform much poorly and hence we can conclude that NB model’s performance is much better than KNN in this scenario.

## Model Tuning, Bagging (Random Forest should be applied for Bagging) and Boosting.

Model tuning relates to measures taken to increase the performance of models. We can perform model tuning by running the various machine learning models through a grid search, which will run the models for various hyper parameters and select the best combination based on the training dataset and does cross validation towards resampling of data to ensure the model performs well with unseen data.

We will start with gridsearch on the 4 models applied on our datasets( in section 1.4 and 1.5), which are – Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbor and Naïve Bayes.

The Naïve Bayes model, which is based on Bayes theory, has no hyper parameters which can be iterated and hence for Naïve bayes model, we will not be running the Grid search, and refer to the same model created in section 1.5.

#### Logistic Regression (LGR):

The hyperparameters and associated values considered in grid search for logistic regression are given below:

* solver – liblinear/lbfgs

This refers to the algorithm which is to be used for optimizing the model. ‘lbfgs’ is the default and is a good first option Generally for smaller datasets (like in our case) liblinear is also considered ideal. Hence we will iterate the model through these 2 options.

* tol – 0.1, 0.01,…, 0.000001

tolerance refers to the value at which optimization can stop. The default value is set at .0001. We will iterate from 0.1 to 0.000001

* max\_iter – 25,50,100, …,1000, 10000

max\_iter refers to the maximum number of iterations required for the solver to converge. Default value is 100 and we will iterate between 25 to 10000.

After running gridsearch on the logistic regression model, with above parameters (cross validation set at 10), the best hyper-parameters were found to be {'max\_iter': 25, 'solver': 'liblinear', 'tol': 0.1}.

Based on the above model, the accuracy for train/test dataset is:

The accuracy of Logistic Regression model tuned by Gridsearch on training set is 0.8406747891283973.

The accuracy of Logistic Regression model tuned by Gridsearch on testing set is 0.8231441048034934.

We can see that there is slight overfitting issue as accuracy of training data set is slightly higher than test dataset. But overall the model’s performance is satisfactory.

Let’s look at the feature importance:

In terms of absolute magnitude, the top 3 features in descending order are –

Hague, Europe, Blair

Text

Description automatically generated

Table-1.35 Feature Importance – Logistic Regression

#### Linear Discriminant Analysis (LDA):

The hyperparameters and associated values considered in grid search for LDA are given below:

* solver – svd/lsqr/eigen

This refers to the algorithm which is to be used for optimizing the model. ‘svd’ is the default and is a good first option as it does not calculate covariance matrix . lsqr (least squares solution) s also an efficient algorithm. eigen solver is based on the optimization of the between class variance to within class variance ratio. Hence we will iterate the model through these 3 options.

* tol – 0.1, 0.01,…, 0.000001

tolerance refers to the value at which optimization can stop. The default value is set at .0001. We will iterate from 0.1 to 0.000001

After running gridsearch on the LDAmodel, with above parameters (cross validation set at 10), the best hyper-parameters were found to be {'solver': 'svd', 'tol': 0.1}.

Based on the above model, the accuracy for train/test dataset is:

The accuracy of LDA model tuned by Gridsearch on training set is 0.8369259606373008.

The accuracy of LDA model tuned by Gridsearch on testing set is 0.8187772925764192.

We can see that there is slight overfitting issue as accuracy of training data set is slightly higher than test dataset. But overall the model’s performance is satisfactory.

Let’s look at the feature importance:

In terms of absolute magnitude, the top 3 features in descending order are –

Hague, Europe, Blair

Text

Description automatically generated

Table-1.36 Feature Importance – LDA

#### K-Nearest Neighbor (KNN):

The hyperparameters and associated values considered in grid search for KNN are given below:

* weights – uniform/distance

This refers to the different kind of weights to be associated with the features. In uniform all points are given equal weights where as in distance based weights, a weight is assigned to points related to inverse of their distance. Hence we will iterate the model through these 2 options.

* n\_neighbors – 3, 5, 7…, 21, 23

This refers to the number of neighbors to be considered. We will iterate from 3 to 23.

After running gridsearch on the KNN model, with above parameters (cross validation set at 10), the best hyper-parameters were found to be {'n\_neighbors': 15, 'weights': 'uniform'}.

Based on the above model, the accuracy for train/test dataset is:

The accuracy of KNN model tuned by Gridsearch on training set is 0.8359887535145267.

The accuracy of KNN model tuned by Gridsearch on testing set is 0.8253275109170306.

We can see that there is slight overfitting issue as accuracy of training data set is slightly higher than test dataset. But overall the model’s performance is satisfactory.

#### Bagging – using Random Forest:

Bagging is an ensemble technique typically used with decision trees/random forest. The base classifier runs in parallel against a intermediate training set which is randomly sampled with replacement against the actual training set. We will be using Random Forest classifier as the base estimator for Bagging classifier.

The hyperparameters and associated values considered in grid search for Random Forest classifier are given below:

* base\_estimator\_\_criterion – gini/entropy

This refers to the algorithm which is to be used for optimizing the model. ‘svd’ is the default and is a good first option as it does not calculate covariance matrix . lsqr (least squares solution) s also an efficient algorithm. eigen solver is based on the optimization of the between class variance to within class variance ratio. Hence we will iterate the model through these 3 options.

* base\_estimator\_\_min\_samples\_leaf – 10, 15, 25

This refers to the minimum number of samples required to be at leaf node. Generally we consider 1% of the dataset as this value. Since we have about ~1000 observations in training dataset, we can set the parameter value at 10 and then iterate for 15 and 25.

* base\_estimator\_\_min\_samples\_split – 30, 45, 75

This refers to the minimum number of samples required to split an internal node. Generally we start this value as 3 times the min\_smaples leaf. Hence we will iterate through 30, 45 and 75.

* base\_estimator\_\_max\_depth – 10, 15, 20

This refers to the maximum depth of the tree. We will iterate through 10,15 and 20.

After running gridsearch on the bagging model, with above parameters (cross validation set at 10), the best hyper-parameters were found to be {criterion='entropy', max\_depth=10, min\_samples\_leaf=15,

min\_samples\_split=30}.

Based on the above model, the accuracy for train/test dataset is:

The accuracy of bagging model tuned by Gridsearch on training set is 0.8416119962511716.

The accuracy of bagging model tuned by Gridsearch on testing set is 0.8209606986899564.

We can see that there is slight overfitting issue as accuracy of training data set is slightly higher than test dataset. But overall the model’s performance is satisfactory.

Let’s look at the feature importance:

In terms of absolute magnitude, the top 3 features in descending order are –

Europe, Hague, Blair

Text

Description automatically generated

Table-1.37 Feature Importance – Bagging

#### Boosting – Adaboosting:

Boosting is an ensemble modeling technique that attempts to build a strong classifier from a number of weak classifiers. Boosting creates a model by using weak models in series. First, a model is built from the training data and subsequently the second model is built which tries to correct the errors present in the first model and this process continues until either the complete training data set is predicted correctly or the maximum number of models are added.

Adaboosting or adaptive boosting is a boosting technique that combines multiple weak classifiers into a strong classifier.

The hyperparameters and associated values considered in grid search for Adaboosting classifier are given below:

* algorithm – SAMME/SAMME.R

The SAMME and SAMME.R algorithms are multiclass Adaboost functions. Hence we will iterate the model through these 3 options.

* n\_estimators – 75, 100, 125

This refers to the number of models to iteratively learn. Default is 50, we will iterate through 75, 100 and 125.

After running gridsearch on the bagging model, with above parameters (cross validation set at 5), the best hyper-parameters were found to be {'algorithm': 'SAMME', 'n\_estimators': 100}.

Based on the above model, the accuracy for train/test dataset is:

The accuracy of Adaboost model tuned by Gridsearch on training set is 0.8481724461105904.

The accuracy of Adaboost model tuned by Gridsearch on testing set is 0.8144104803493449.

We can see that there is slight overfitting issue as accuracy of training data set is slightly higher than test dataset. But overall the model’s performance is satisfactory.

Let’s look at the feature importance:

In terms of absolute magnitude, the top 3 features in descending order are –

Europe, Hague, Blair

Text

Description automatically generated

Table-1.38 Feature Importance – Adaboosting

#### Boosting – Gradient Boosting:

Boosting is an ensemble modeling technique that attempts to build a strong classifier from a number of weak classifiers. Gradient boosting gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.

The hyperparameters and associated values considered in grid search for gradient boosting classifier are given below:

* loss – deviance/ exponential

The loss function to be optimized. 'deviance' refers to deviance (= logistic regression) for classification with probabilistic outputs. For loss 'exponential' gradient boosting recovers the AdaBoost algorithm.

* criterion – friedman\_mse/mse

The function to measure the quality of a split. Supported criteria are 'friedman\_mse' for the mean squared error with improvement score by Friedman, 'mse' for mean squared error.

* min\_samples\_leaf - 8, 10, 12

This refers to the minimum number of samples required to be at leaf node. Generally we consider 1% of the dataset as this value. Since we have about ~1000 observations in training dataset, we can set the parameter value at 10 and then iterate for 8 and 12.

* min\_samples\_split - 25, 30, 35

This refers to the minimum number of samples required to split an internal node. Generally we start this value as 3 times the min\_samples leaf. Hence we will iterate through 25, 30 and 35.

* max\_depth - 5, 10, 15

This refers to the maximum depth of the tree. We will iterate through 5,10 and 15.

After running gridsearch on the gradient boosting model, with above parameters (cross validation set at 5), the best hyper-parameters were found to be {'criterion': 'friedman\_mse', 'loss': 'exponential', 'max\_depth': 5, 'min\_samples\_leaf': 12, 'min\_samples\_split': 35}.

Based on the above model, the accuracy for train/test dataset is:

The accuracy of Gradient boosting model tuned by Gridsearch on training set is 0.9268978444236177.

The accuracy of Gradient boosting model tuned by Gridsearch on testing set is 0.8231441048034934.

We can see that there is significant overfitting issue as accuracy of training data set is considerably higher than test dataset. Hence gradient boosting model’s performance is not apt in this scenario.

Let’s look at the feature importance:

In terms of absolute magnitude, the top 3 features in descending order are –

Hague, Europe, age

Text

Description automatically generated

Table-1.39 Feature Importance – Gradient boosting

## Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.

Lets check the performance metrics of all the models:

#### Logistic Regression (LGR):

* Accuracy:

The accuracy of Logistic regression model on training set is 0.8406747891283973.

The accuracy of Logistic regression model on testing set is 0.8231441048034934

* Confusion matrix:

  Table-1.40 Confusion Matrix: Logistic Regression – Training Data Table-1.41 Confusion Matrix: Logistic Regression – Testing Data

* ROC\_AUC score:

The ROC\_AUC score for Logistic Regression model on training data is 0.8893389886074912.

The ROC\_AUC score for Logistic Regression model on testing data is 0.8820825515947467.

* The ROC curve is given below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure-1.36 ROC Curve: LGR – Training Data Figure-1.37 ROC Curve: LGR – Testing Data

* Classification Report:

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Table-1.42 Classification Report: LGR – Training Data Table-1.43 Confusion Matrix: LGR– Testing Data

#### Linear Discriminant Analysis (LDA):

* Accuracy:

The accuracy of LDA model on training set is 0.8369259606373008.

The accuracy of LDA model on testing set is 0.8187772925764192.

* Confusion matrix:

  Table-1.44 Confusion Matrix: LDA – Training Data Table-1.45 Confusion Matrix: LDA – Testing Data

* ROC\_AUC score:

The ROC\_AUC score for LDA model (Gridsearch) on training data is 0.8892242439144332.

The ROC\_AUC score for LDA model (Gridsearch) on testing data is 0.8837711069418386.

* The ROC curve is given below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure-1.38 ROC Curve: LDA – Training Data Figure-1.39 ROC Curve: LDA – Testing Data

* Classification Report:

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Table-1.46 Classification Report: LDA – Training Data Table-1.47 Confusion Matrix: LDA– Testing Data

#### K-Nearest Neighbor (KNN):

* Accuracy:

The accuracy of KNN model on training set is 0.8359887535145267.

The accuracy of KNN model on testing set is 0.8253275109170306.

* Confusion matrix:

  Table-1.48 Confusion Matrix: KNN – Training Data Table-1.49 Confusion Matrix:KNN– Testing Data

* ROC\_AUC score:

The ROC\_AUC score for KNN model (Gridsearch) on training data is 0.9071346610933531.

The ROC\_AUC score for KNN model (Gridsearch) on testing data is 0.89375.

* The ROC curve is given below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure-1.40 ROC Curve: KNN– Training Data Figure-1.41 ROC Curve: KNN – Testing Data

* Classification Report:

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Table-1.50 Classification Report:KNN– Training Data Table-1.51 Confusion Matrix: KNN– Testing Data

#### Naïve Bayes (NB):

* Accuracy:

The accuracy of NB model on training data is 0.8331771321462043.

The accuracy of NB model on testing set is 0.8253275109170306.

* Confusion matrix:

Text

Description automatically generated 

Table-1.52 Confusion Matrix:NB– Training Data Table-1.53 Confusion Matrix: NB– Testing Data

* ROC\_AUC score:

The ROC\_AUC score for NB model on training data is 0.8864703712810426.

The ROC\_AUC score for KNN model on testing data is 0.8845450281425891.

* The ROC curve is given below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure-1.42 ROC Curve: NB – Training Data Figure-1.43 ROC Curve: NB – Testing Data

* Classification Report:

Calendar

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Table-1.54 Classification Report: NB– Training Data Table-1.55 Confusion Matrix: NB – Testing Data

#### Bagging with Random Forest (BAG):

* Accuracy:

The accuracy of Bagging model on training set is 0.8416119962511716.

The accuracy of bagging model on testing set is 0.8209606986899564.

* Confusion matrix:

  Table-1.56 Confusion Matrix: BAG – Training Data Table-1.57 Confusion Matrix: BAG– Testing Data

* ROC\_AUC score:

The ROC\_AUC score for bagging model (Gridsearch) on training data is 0.9078046881403162.

The ROC\_AUC score for bagging model (Gridsearch) on testing data is 0.8954268292682925.

* The ROC curve is given below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure-1.44 ROC Curve: BAG– Training Data Figure-1.45 ROC Curve: BAG – Testing Data

* Classification Report:

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Table-1.58 Classification Report: BAG– Training Data Table-1.59 Confusion Matrix: BAG– Testing Data

#### Boosting - Adaboost(ADB):

* Accuracy:

The accuracy of Boosting - Adaboost model on training set is 0.8481724461105904.

The accuracy of Boosting - Adaboost model on testing set is 0.8144104803493449.

* Confusion matrix:

  Table-1.60 Confusion Matrix: ADB – Training Data Table-1.61 Confusion Matrix: ADB– Testing Data

* ROC\_AUC score:

The ROC\_AUC score for Boosting - Adaboost model on training data is 0.9023850504057046

The ROC\_AUC score for Boosting - Adaboost model on testing data is 0.8851430581613507.

* The ROC curve is given below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure-1.46 ROC Curve: ADB– Training Data Figure-1.47 ROC Curve: ADB – Testing Data

* Classification Report:

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Table-1.62 Classification Report: ADB– Training Data Table-1.63 Confusion Matrix: ADB– Testing Data

#### Gradient Boosting(GDB):

* Accuracy:

The accuracy of Gradient Boosting model on training set is 0.9268978444236177.

The accuracy of Gradient Boosting model on testing set is 0.8231441048034934.

* Confusion matrix:

  Table-1.64 Confusion Matrix: GDB – Training Data Table-1.65 Confusion Matrix: GDB– Testing Data

* ROC\_AUC score:

The ROC\_AUC score for Gradient Boosting model on training data is 0.9809175477419884.

The ROC\_AUC score for Gradient Boosting model on testing data is 0.8914165103189493.

* The ROC curve is given below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Figure-1.48 ROC Curve: GDB– Training Data Figure-1.49 ROC Curve: GDB – Testing Data

* Classification Report:

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

Table-1.66 Classification Report: GDB– Training Data Table-1.67 Confusion Matrix: GDB– Testing Data

Lets check all the model’s performance:

Graphical user interface

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with low confidence

\ Table-1.68 Model comparison

Gradient boost model (GDB\_train and GDB\_test) is overfitted, hence we will not consider it for our pick of the optimal model for the concerned dataset.

All the models are slightly overfitted.

For accuracy we can see that Naïve Bayes has scored highest for test dataset.

In precision also we have Naïve Bayes which has scored highest for test dataset.

For recall we can see that Bagging with Random Forest has scored the highest.

For f1-score we can see that logistic regression and Bagging with Random Forest has scored the highest.

For roc\_auc\_score we can see that Bagging with Random Forest has scored the highest.

From all the parameters we can consider Naïve bayes model to performing optimally.

## Based on these predictions, what are the insights?

#### Business Insights:

* Irrespective of the economic condition of the voter’s household, majority of voters have voted for the Labour party. The vote % between Labour and Conservative party is comparable for 1 and 2 economic condition household, but for 3,4 and 5 Labour party enjoys a clear majority with 4 having the maximum vote difference.
* Majority of the voters who have higher assessment of current national economic condition (3,4 and 5) have voted for Labour party, but majority of those voters who had lower assessment (1 and 2) have voted for the Conservative party. The lead enjoyed by Conservative party is minimal for 1 and 2, whereas the majority enjoyed by Labour party for voters assessing current national economic condition (3,4 and 5) is at least 100% or more.
* Irrespective of the voter’s knowledge of the position of the parties on European integration, majority of the voters in each class have voted for the Labour party and the party enjoys a comfortable margin over Conservative party at least 70% or more
* Irrespective of the voter’s gender, Labour party enjoys a clear majority over Conservative party at least 100% or more.
* Conservative party enjoys a majority among voters with high Eurosceptic sentiment (9,10 and 11). For all other assessment values Labor party enjoys a clear majority. We can see that the share of voters for Conservative party seems to be increasing with Eurosceptic sentiment, whereas it seems that for voters who have chosen Labour party seems not to consider European integration as a factor
* On an average with increase in age of the voter, the assessment of the Labour leader Blair seems to increase.
* Age of the voter does not seem to be a factor influencing the assessment of the Conservative leader Hague
* Voters who have lower assessments for Blair (1 and 2) have voted for Conservative party. But interestingly voters do not seem to have a neutral stand towards Blaire (3). Majority of voters who have high assessment of Blair (4 and 5) have voted for Labour party.
* Majority of the voters who have lower assessments for Hague(1 and 2) have voted for Labour party. Majority of voters who have high assessment of Hague (4 and 5) have voted for Conservative party
* Irrespective of the gender, majority of the voters have a higher assessment of Blair and a lower assessment of Hague.
* Majority of voters who assessed economic\_cond\_household as 3 and above (which is majority of the sample) have assessed Blaire highly
* Majority of voters who assessed economic\_cond\_household as 3 and above (which is majority of the sample) have assessed Hague low (1 and 2), whereas below 3 we can see that majority of voters have assessed Hague highly
* Irrespective of the household economic condition majority of the voters are Eurosceptic
* Irrespective of the national economic condition assessment, majority of the voters in each class are Eurosceptic

#### Business Recommendations:

* Business needs to ensure that sampling of the populations is corresponding to the demographics of the province. We need to sample for all the provinces and then extrapolate for being better in predicting poll results.
* Younger population and first-time voter’s data seems to be missing from the sample, and this needs to be accounted for.

# Problem 2

In this particular project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

President Franklin D. Roosevelt in 1941

President John F. Kennedy in 1961

President Richard Nixon in 1973

## Find the number of characters, words, and sentences for the mentioned documents.

No. of characters in Roosevelt's speech : 7571

No. of characters in Kennedy's speech : 7618

No. of characters in Nixon's speech : 9991

No. of words in Roosevelt's speech : 1526

No. of words in Kennedy's speech : 1543

No. of words in Nixon's speech : 2006

No. of sentences in Roosevelt's speech : 68

No. of sentences in Kennedy's speech : 52

No. of sentences in Nixon's speech : 68

## Remove all the stopwords from all three speeches.

From nltk corpus we will retain the stop words for ‘english’ language. We will also consider the punctuation marks in the stop words.

We have a total of 211 stop words and punctuation marks. Some of the stop words can be seen below:



Table-2.1 Stop words: Sample

Let us check the statistics of stop words, and sample of stop words identified to be removed from the 3 speeches:

A picture containing graphical user interface

Description automatically generated

Table-2.2 Stop words statistics

Let us check the statistics of the speech after the stop words have been removed.

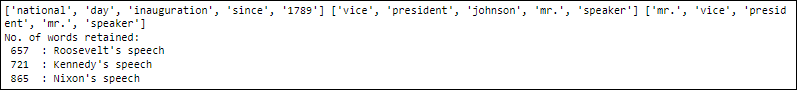


Table-2.3 Speech statistics after stop words removal

We will not perform stemming or lemmatization as this will transform the word and we need the actual top words spoken by the president’s in their inaugural speech.

## Which word occurs the most number of times in his inaugural address for each president? Mention the top three words.

We will use nltk’s frequency distribution, to get the most common 3 words:

Text

Description automatically generated

Table-2.4 Top 3 words of each speech

## Plot the word cloud of each of the three speeches. (after removing the stopwords).

* Lets look at the world cloud of Roosevelt’s speech:

A picture containing text, newspaper

Description automatically generated

Figure-2.1 Word cloud: Roosevelt’s speech

* Lets look at the world cloud of Kennedy’s speech:

Text

Description automatically generated

Figure-2.2 Word cloud: Kennedy’s speech

* Lets look at the world cloud of Nixon’s speech:

Text

Description automatically generated

Figure-2.3 Word cloud: Nixon’s speech

## THE END